[0086] The commonest class of error-correcting codes are linear error-correcting codes. Almost all of the error-correcting codes presently used in practice are linear. It is convenient, although not necessary, to choose the decoding function of a linear error-correcting code for use in embodiments of the present invention. One property of linear error-correcting codes that is useful in a number of applications is that it is easy to select a codeword c uniformly at random from the set of codewords C.

[0087] It should be noted, however, that an error-correcting code traditionally involves changing a message m to a codeword c before transmission **30**. In some situations, however, the translation function g cannot be applied effectively. For instance, when the message m itself contains errors, generating redundancy is problematic. The errors in the message m may well be propagated and reinforced by the redundancy in the corresponding codeword c. This situation exists in the case of a secret pattern that comprises a sequence of discrete graphical choices. It may be difficult for a user to make or repeat discrete graphical choices on a graphical interface without errors; accordingly, a sequence of values that corresponds to a secret pattern should be considered a message m that includes errors. Thus, embodiments of the present invention do not use error-correcting codes in the traditional way.

[0088] Embodiments of the present invention use the decoding function f of an error-correcting code to relate a value, which corresponds to a discrete graphical choice, to a codeword c. In some embodiments, the value is treated as a corrupted codeword i in an error-correcting code. In such embodiments, the decoding function f decodes the value into a codeword c as if the value were a corrupted codeword i.

[0089] Embodiments of the invention do not make use of the translation function g or the reverse translation function g^{-1} of the error-correcting code. In consequence, such embodiments do not map a message m from the message space **10** to a codeword c from the set of codewords C in codeword space **20**. Nor do such embodiments map a codeword c from the set of codewords C in codeword space **20** back to a message m from the message space **10**. In fact, such embodiments do not use the message space **10** at all.

[0090] The enrollment process illustrated in **FIG. 2** includes selecting a codeword c for a value (STEP **230**). The value is an n-bit string that corresponds to a discrete graphical choice. The graphical choice may be any sort of discrete input to a graphical interface, such as the selection of a region, an area, or a point on the graphical interface. The codeword c that is selected for a value is also an n-bit string. In some embodiments, STEP **230** involves applying a decoding function f of an error-correcting code to a value. In one such embodiment, the decoding function f is part of a linear error-correcting code.

[0091] In one embodiment, the error-correcting code has a dimension d in which codewords are of the form <Ra_**1**, Ra_**2**, . . . , Ra_d> such that a_i is an integer and R is a real-valued code parameter and the decoding function f as applied to vector <x_**1**, x_**2**, . . . , x_d> simply rounds each element x_i to the integer a_iR that is closest. In a related embodiment, where there are ambiguities, a deterministic or randomized tie-breaking algorithm is used, or both possibilities are checked.

[0092] **FIG. 9** illustrates a geometric analogy for selecting a codeword c for a value x (STEP **230**), according to one embodiment of the present invention. The set of codewords C are represented as the set of points c_1, c_2, c_3, and c_4 on the u-v plane; mathematically expressed as $C=\{c_1, c_2, c_3, c_4\}$. The value x is represented as a point on the u-v plane in **FIG. 9** with the coordinates (**30**, **595**). The decoding function f associated with this example, but not shown, selects the codeword c within the set of codewords C that is nearest to the input. In comparison, the decoding function f in its normal use as part of an error-correcting code would select the codeword c that is nearest to the corrupted codeword i. Accordingly, since in this example the input to the decoding function is the value x, the decoding function f selects codeword c_3, the codeword nearest to the value x. This process is expressed mathematically as $f(x)=c_3$.

[0093] Note that **FIG. 9** illustrates a geometric analogy of an embodiment in which the decoding function f has no minimum distance in contravention to the usual case. A decoding function f that uses unconstrained codewords selects the codeword c nearest the value x without limit on its distance between the value x and the nearest codeword c. Such a decoding function f will not always be successful in selecting a codeword c for a value x because, for example, a value x may be equidistant to more than one codeword.

[0094] **FIG. 10** illustrates a geometric analogy of an embodiment in which the decoding function f selects the codeword c nearest to the value x provided that the distance between the value x and the nearest codeword c falls within the minimum distance of the error-correcting code. In such an embodiment, a decoding function f that uses constrained codewords is used in STEP **230**. The dotted line circles surrounding each of the codewords C $\{c_1, c_2, c_3, c_4\}$ represent the boundaries of the neighborhood of values for which the decoding function f will select the included codeword c. The decoding function f will not select the included codeword c for any value outside the dotted line circle that surrounds the codeword c, even if the value outside the dotted line circle is closer to the enclosed codeword c than to any other codeword. For example, the value x in **FIG. 10** does not fall within the boundaries of an area that will map to any codeword c. Accordingly, the decoding function f may output ϕ.

[0095] In either of the foregoing embodiments, the amount of information about the value x that is contained in the corresponding codeword c depends on the number of codewords, or the size of the set of codewords C. The larger the set of codewords, the more information that a codeword contains about its associated value x.

[0096] A comparison of **FIG. 11**A and **FIG. 11B** illustrates this concept through another geometric analogy. **FIG. 11A** shows a value x and a set of one codeword $C=\{c\}$, both as points on a plane. A decoding function f will select the codeword c nearest to the value x in **FIG. 11A**. Since there is a single codeword c in the set of codewords C in **FIG. 11**A, the decoding function f will select the codeword c for the value x. Accordingly, knowing that the decoding function selected codeword c for the value x in **FIG. 11A** provides no information about the true location of the value x on the plane. In comparison, **FIG. 11B** shows a value x and a set of four codewords, all as points on a plane. A decoding function f will select the nearest codeword c_1 for the value x in **FIG. 11B**. Since there are four codewords in the set of codewords $C=\{c_1, c_2, c_3, c_4\}$ in **FIG. 11**B and the value x is